KaliDoS, un outil de vérification des règles de catalogage bibliographique

Nuria Pastor Martinez, responsable du Pôle Données et Signalement (SCD de l'UCBL)

Fabien Duchateau, enseignant-chercheur (UCBL et laboratoire LIRIS UMR5205)

Chaque année, l'Université Claude Bernard Lyon 1 (UCBL) signale, en moyenne, 23 000 nouveaux titres dans le Sudoc. Environ 5 000 nécessitent la création d'une notice bibliographique et sont, pour la plupart, des 'unicas' (documents possédés uniquement par le SCD de l'UCBL, par exemple des thèses, des mémoires, des fonds anciens numérisés). La qualité de ces notices bibliographiques est primordiale pour garantir l'accès aux ressources. Pourtant, peu d'applications existent pour faciliter le contrôle qualité et elles sont non exhaustives voire obsolètes du fait de l'évolution des règles de catalogage. En collaboration avec le SCD, un groupe de six étudiant.e.s du Master 2 "technologies de l'information et du web" de l'UCBL a donc été chargé de développer une application, appelée KaliDoS (Qualité des Données du Sudoc), permettant de détecter, parmi un lot de notices, celles qui ne respectent pas un ensemble de règles. La modélisation des règles à appliquer pour valider (ou non) les notices constituait un des enjeux majeurs de ce projet.

Modèle de règles

L'un des défis pour l'implémentation de KaliDoS réside dans la gestion des règles, que ce soit pour leur représentation ou leur utilisation. Différents types de règles ont été identifiés. Cette catégorisation permet de rendre générique la définition de ces règles, et donc d'en ajouter plus facilement.

Voici un aperçu de ces catégories :

Catégorie	Définition
Comptage	Vérifie que le nombre de sous-zones d'une zone soit égal au nombre de sous-zones d'une autre zone. *Exemple : plusieurs sous-zones 101$d nécessitent autant de sous-zones 330$a.*
Dépendance	Compare la valeur de deux zones contenues dans un même PPN. *Exemple : les 4 premiers caractères de la sous-zone 029$b doivent être égaux ceux de la sous-zone 328$d.* Les opérateurs de comparaison sont l'égalité, l'inégalité et les relations inférieure et supérieure (strictes ou non).
IdRef	Vérifie si une zone soumise à autorité est valide en interrogeant un référentiel externe. *Exemple : une notice contenant une sous-zone 606$2 avec la valeur "rameau" indique qu'il faut vérifier si le PPN enregistré en sous-zone 606$3 renvoie bien sûr une autorité Rameau, via l'analyse de sa zone de contrôle 008.*
Matching	Vérifie si la valeur d'une sous-zone correspond à une expression régulière. *Exemple : la sous-zone 339$d ne doit pas contenir "Année de mise en ligne".*

Ordonnancement	Vérifie qu'une liste de zones possédant le même tag soit triée par indicateur. *Exemple : s'il y a plusieurs zones 214, elles doivent être ordonnées selon l'indicateur "ind2".*
Précédence	Vérifie, au sein d'une zone, qu'une sous-zone est bien précédée par une autre. *Exemple : une sous-zone 608$a doit être précédée d'une 608$3.*
Structure	Vérifie la présence d'une zone, d'une sous-zone ou d'indicateur. *Exemple : la zone 328 doit contenir un indicateur "ind2" avec la valeur 0.* Différents types de vérifications sont proposés, par exemple l'absence ou la présence d'une zone dans la notice, l'obligation de fournir une valeur à une zone, ou la présence d'un code pour une zone.
DépendanceConditionnelle	Vérifie une règle de dépendance si la notice répond à une condition particulière. *Exemple : s'il y a une zone 455, alors la date en sous-zone 455$d doit correspondre à la date indiquée en zone 100 position 13-16.*
MatchingConditionnel	Vérifie une règle de matching si la notice répond à une condition particulière. *Exemple : si la sous-zone 328$z ne mentionne pas "Reproduction de", alors il ne doit pas y avoir de zone 608 $3027253139Thèses et écrits académiques.*
StructureConditionnel	Vérifie une règle de structure si la notice répond à une condition particulière. *Exemple : si la zone 008 commence par Oa, une zone 304 doit être présente.*

Jeu de règles

Chaque type de document est vérifié selon un jeu de règles qui lui est propre. Par défaut, cinq jeux de règles sont créés avec KaliDoS (CITER ICI LES JEUX, dont un jeu général), et une interface graphique permet d'éditer les jeux de règles ainsi que le contenu de chaque règle. Si la majorité des règles sont communes à tous les établissements du réseau, il peut exister des règles locales, qui n'ont donc pas vocation à être utilisées ailleurs (par exemple, pour les thèses, le libellé normalisé de chaque université en sous-zone 328$e ou le point d'accès à l'établissement de soutenance en sous-zone 711$3).
Les règles sont stockées au format JSON (un fichier par jeu). Plusieurs exemples sont disponibles dans la documentation sur les règles. Ci-dessous une règle qui vérifie que la zone 230$a, qui stocke la taille d'une ressource électronique, ne contient pas l'unité 'Mo'.

```
{
    "number": 230,                                          // numéro du champ à vérifier
    "code": "a",                                            // code du sous-champ à vérifier
    "regex": "(?:(?!Mo).)+",                                // expression régulière à vérifier
    "message": "Zone 230 : corriger le poids en Mo",        // message à afficher en cas de violation
    "index": 39                                             // identifiant unique de la règle
}
```

Limitation du modèle

La gestion des règles est limitée sur trois points.
Tout d'abord, la distinction entre règle spécifique locale et règle collective n'est pas stricte : une application utilisée par plusieurs établissements se doit donc de marquer cette différence afin de ne partager que les règles communes à tout un réseau.
De plus, il n'y a pas de notion d'héritage, i.e., un jeu de règles spécifique ne peut pas hériter des règles d'un jeu général. Ce choix se justifie par le fait que les règles évoluent peu fréquemment, mais une application nationale devrait permettre cet héritage afin de faciliter la réutilisation et le partage des règles entre établissements.
Enfin, il n'y a aucun opérateur pour combiner des règles : un type de règle qui doit vérifier à la fois une condition et la structure de la notice a donc été définie au lieu de combiner une règle conditionnelle et une structurelle. Il y aurait donc matière à repenser ce modèle de règles afin de permettre de combiner des règles existantes.

Conclusion

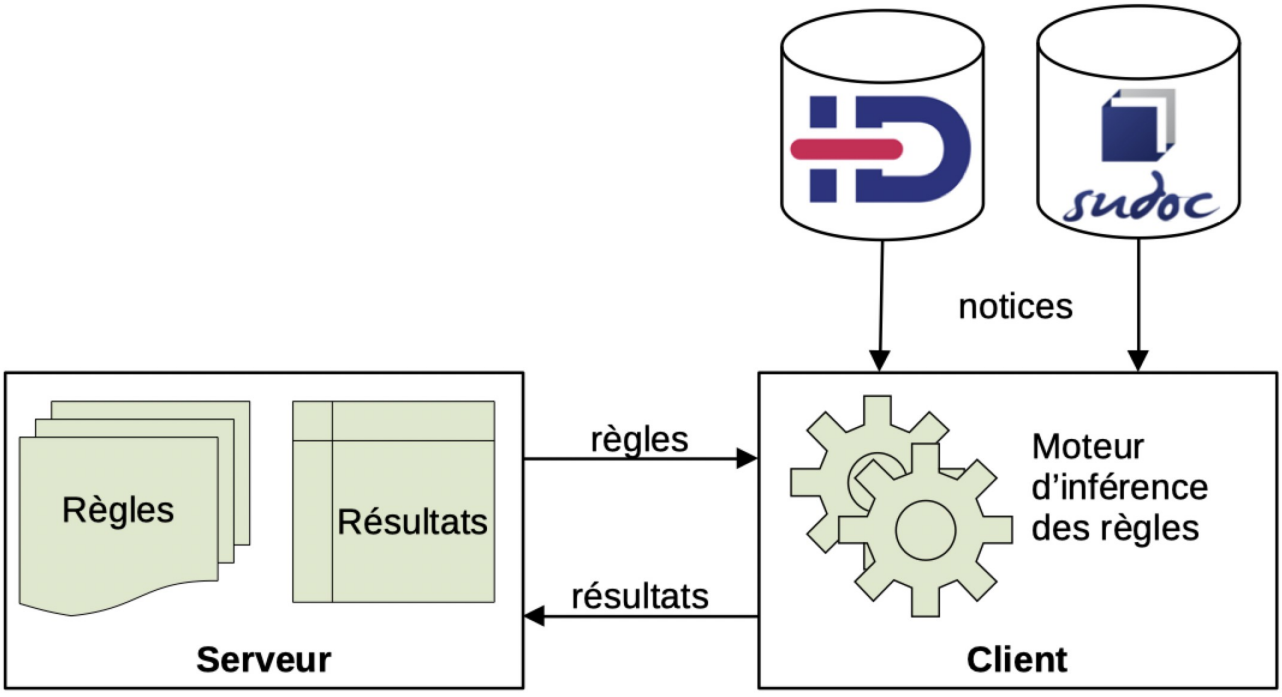
Une application dédiée au contrôle qualité des notices bibliographiques doit inclure un modèle de règles extensible. Celui de KaliDoS répond bien aux besoins actuels, mais pourra nécessiter des adaptations s'il était utilisé par un réseau d'établissements. Dans un second billet, nous décrirons l'architecture et les interfaces de KaliDoS.

KaliDoS, un outil de vérification des règles de catalogage bibliographique (partie 2)

Afin d'améliorer la qualité de son catalogue et de rendre le contrôle qualité des notices bibliographiques plus efficace, le SCD de l'UCBL a souhaité se doter d'un outil de diagnostic, KaliDoS (Qualité des Données du Sudoc). Après une présentation sur la modélisation des règles, nous décrivons dans ce second billet l'architecture de KaliDoS ainsi que les fonctionnalités et les interfaces de l'application.

Architecture de KaliDoS

L'application suit une architecture client-serveur : sur la figure suivante, le serveur stocke les jeux de règles et les résultats dans des fichiers JSON. En plus de la présentation des interfaces, le côté client est en charge d'exécuter le contrôle qualité, après avoir récupéré les notices auprès de deux fournisseurs (IDREF et SUDOC) ainsi que le jeu de règles depuis le serveur. La dockerisation facilite le déploiement de l'application ainsi que son redémarrage en cas d'arrêt critique.

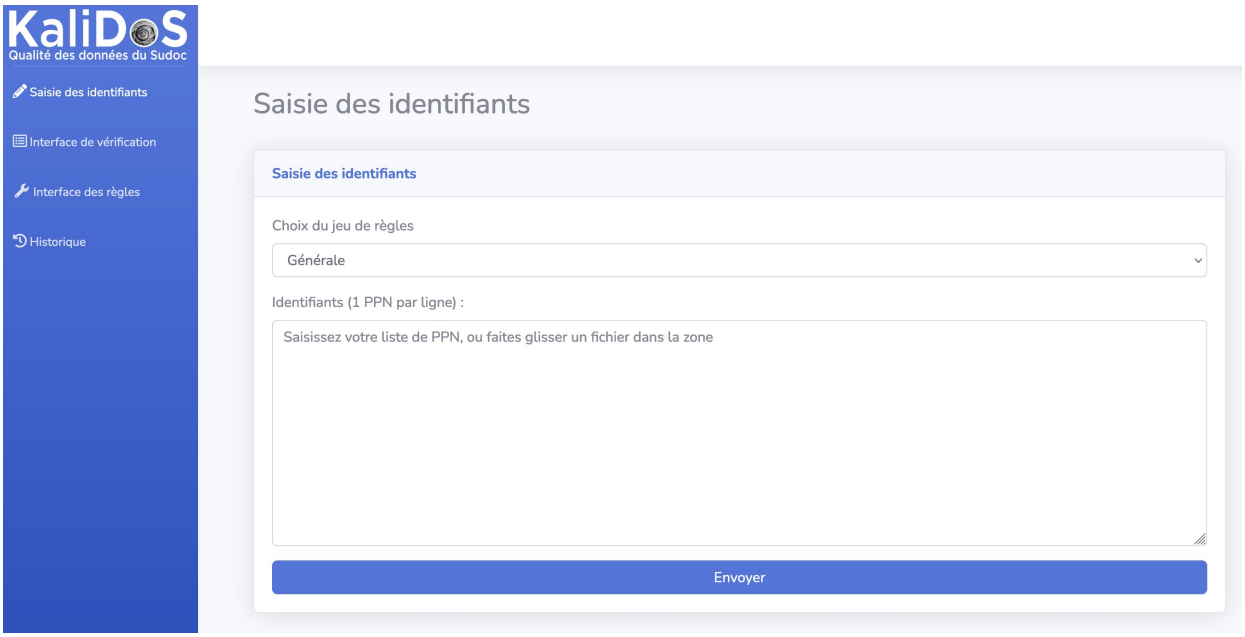


Fonctionnalités et interfaces de KaliDoS

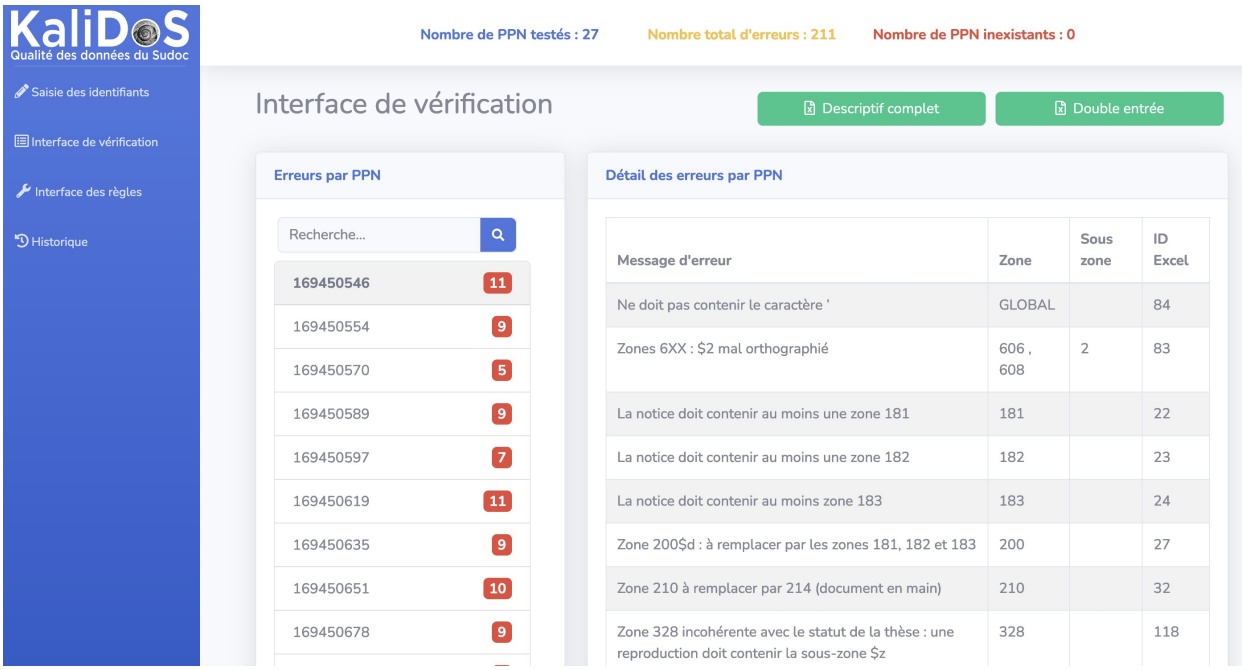
L'application KaliDoS a été développée avec le souci de proposer une interface la plus épurée mais aussi la plus ergonomique possible. Elle est notamment responsive design afin de s'adapter aux différents dispsositifs et permet d'exécuter l'ensemble des règles, d'afficher les résultats et offre l'accès à un éditeur de règles.

Quatre interfaces la composent :

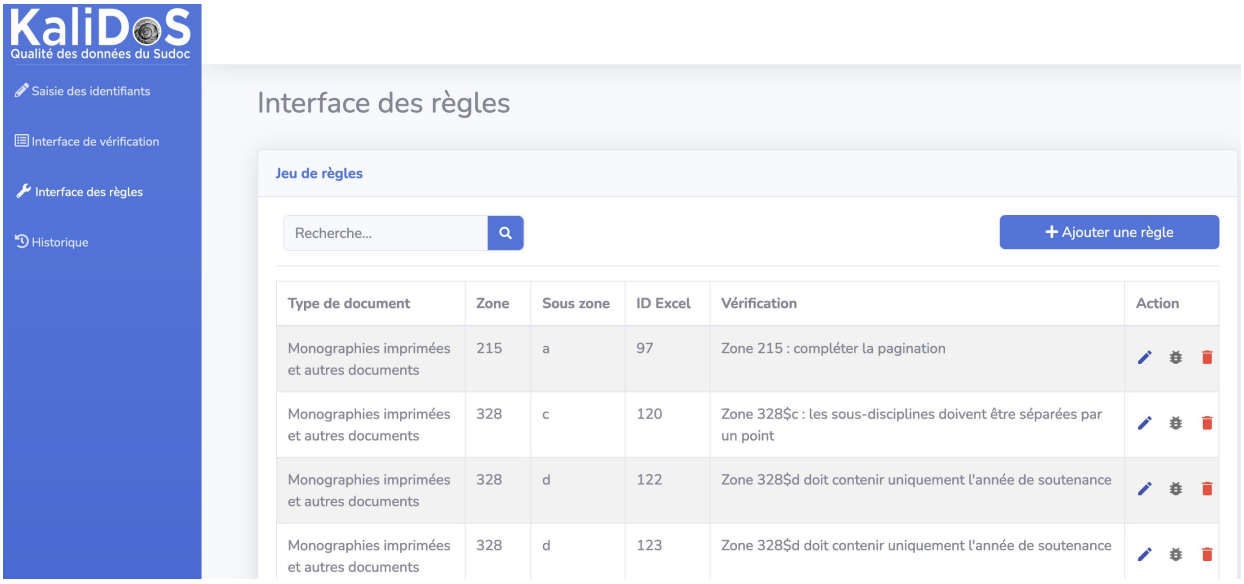
- *Saisie des identifiants* : cette interface permet la saisie d'un ou plusieurs PPN à contrôler, soit par copier-coller, soit en glissant un fichier txt. Avant de lancer la vérification, un menu déroulant permet de choisir le type de règles à appliquer aux notices selon le type de document qu'elles décrivent (voir billet 1 LIEN). Les contrôles s'effectuent par une interrogation de la base XML du Sudoc et d'Idref pour les zones soumises à autorité (mentions de responsabilité et sujets). Elle permet des temps de réponse très courts (500 notices sont analysées en moins d'une minute). Elle rend, par contre, complexe voire impossible certains contrôles du fait des différences entre le format de production dans le Sudoc et le format d'export XML, et surtout par l'absence de certaines données.



- *Interface de vérification* : lorsque l'utilisateur lance un contrôle, il bascule sur l'interface de vérification. Les résultats s'affichent au fur et à mesure de l'analyse et une jauge permet de suivre le pourcentage de notices contrôlées. Une fois le contrôle terminé, celle-ci est remplacée par un résumé du nombre de PPN testés et du nombre d'erreurs identifiées. La liste de notices comportant des erreurs s'affiche à gauche de l'écran (numéro de PPN et nombre d'erreurs) et il est possible de cliquer sur chacune des notices pour afficher, à sa droite, le détail des erreurs détectées. Un export Excel de ces résultats est également possible pour une analyse globale.

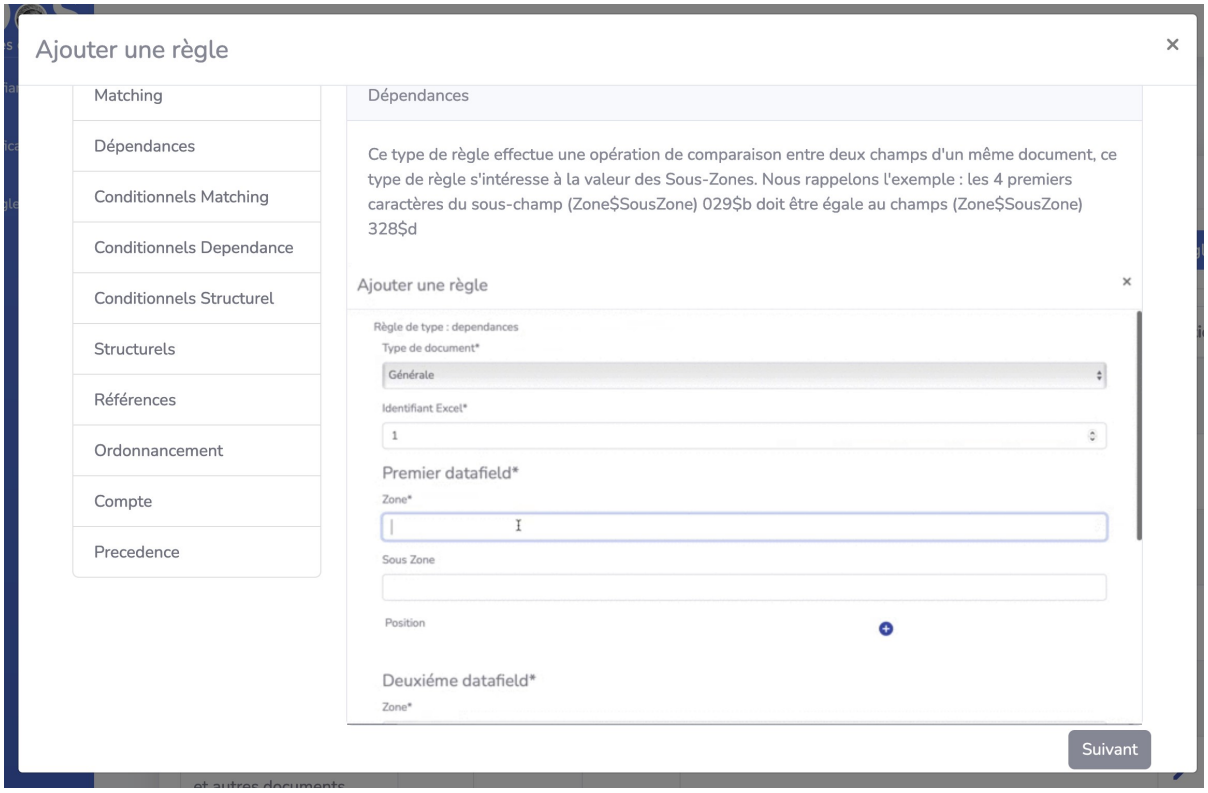


- *Interface des règles* : la véritable plus-value de KaliDoS réside dans son éditeur de règles, résultat d'un travail de modélisation. Il permet d'afficher, de tester et de modifier les règles existantes mais également d'en créer de nouvelles sans avoir de compétences en informatique. Avec les évolutions permanentes des règles et des consignes de catalogage dans le cadre de la Transition bibliographique, cette fonctionnalité s'est révélée essentielle afin de garantir la mise à jour des règles.



Par défaut, toutes les règles s'affichent dans cette interface mais un champ de recherche permet d'affiner l'affichage à partir d'un numéro de zone Unimarc (p. ex., tous les contrôles qui interviennent sur la zone de la collation B215) ou d'un mot clé (p. ex., « rameau »

affichera les contrôles sur les zones B6XX avec une sous-zone $2rameau). Pour l'ajout d'une nouvelle règle, l'utilisateur doit choisir parmi les différents types de règles. Une aide dynamique permet d'expliquer simplement comment doivent être remplis les différents champs de chaque règle.



- *Historique* : il est possible de retrouver l'ensemble de contrôles effectués. Chacun d'eux peut être relancé, par exemple après la correction des notices erronées, ou supprimé de l'historique.

Conclusion et perspectives

L’application KaliDoS répond aux enjeux prévus au départ. Elle est utilisée régulièrement au SCD de l’UCBL depuis février 2021 pour l’identification des notices à corriger mais aussi pour les besoins en formation lorsque des erreurs sont récurrentes. Son éditeur de règles offre à l’application une réelle évolutivité. Il pourrait aussi permettre à un autre établissement d’adapter les règles à ses besoins, ceci d’autant plus que le code est disponible sous licence libre : https://github.com/abes-esr/kalidos.
En dehors des aspects fonctionnels, ce projet a également permis une collaboration enrichissante entre deux services de l’Université et ses étudiants, ce qui n’est malheureusement pas très fréquent. Il a également ouvert la porte à une collaboration avec l’Abes puisque, face au développement de différents outils au sein des établissements du réseau et au besoin croissant des catalogueurs de disposer d’un outil fiable et conforme aux évolutions des règles de catalogage RDA-FR, l’Agence s’est engagé dans le développement d’un outil unique de contrôle bibliographique qui permettra l’accompagnement des catalogueurs du réseau.